DOCUMENT RESUME

ED 297 704 IR 012 655

AUTHOR Hutchinson, Thomas Lloyd
TITLE Surface Description and Motion Control for Animated Three Dimensional Computer Generated Characters.
PUB DATE 86
NOTE 65p.; Master's Thesis, Ohio State University.
PUB TYPE Dissertations/Theses - Master Theses (042) -- Information Analyses (070)

EDRS PRICE MF01/PC03 Plus Postage.
DESCRIPTORS *Animation; Characterization; *Computer Graphics; Computer Software; *Motion; *Production Techniques; *Three Dimensional Aids

ABSTRACT

This study of the relationship of computer technology to character animation focuses on the advantages and constraints of developing three-dimensional characters for computer animation. Three different levels of the complexity involved in animating characters are examined: (1) a three-dimensional computer environment and simple motion within this environment; (2) motion control of rigid monolithic objects, including the principles of acceleration/deceleration, the physical laws of motion, simple squash and stretch, and object attachment and articulated appendages for characters; and (3) shape change techniques for animating characters, including deformations, shape interpolation, growth, fractals, and particle systems. The principles of three-dimensional environment character interaction are also described. Interfacing between different animation software programs is identified as a serious problem for the computer animators, and the development of an industry standard for software is strongly advocated as a first step toward solving this problem. Thirteen references are included. (MES)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED297704

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

X This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

SURFACE DESCRIPTION AND MOTION CONTROL FOR ANIMATED THREE DIMENSIONAL COMPUTER GENERATED CHARACTERS

A Thesis

Presented in Partial Fulfillment of the Requirements
for the Degree Master of Arts

by

Thomas Lloyd Hutchinson, B.S.

The Ohio State University
Summer, 1986

Copyright c 1986
by Thomas L. Hutchinson
All rights resevered

Approved by

Thomas E. Linehan

Advisor
Department of Art Education

IR012455

BEST COPY AVAILABLE

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Thomas L. Hutchinson

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Acknowledgments

I am indebted to my partner, William Sadler, for the support and leadership he provided throughout. I acknowledge the extra effort and time provided by my advisor, Dr. Thomas Linehan, and my personal friends at the Computer Graphics Research Group.

Finally, my special thanks to my Grandfather, George Metcalf, for being an inspiration for me over all these years. And to my mother and sisters, Carroll, Margie, and Beth Hutchinson.

VITA

January 31, 1961.....Born - Montreal, Canada

1984.....B.S. Fine Arts,
School of Arctitecture and
Allied Arts,
University of Oregon,
Eugene, Oregon

Summer 1985.....Instructor, micro-computer
graphics,
Columbus Public Schools,
Columbus, Ohio

1984-Present.....Graduate Research Assistant,
Computer Graphics Research Group,
Ohio State University,
Columbus, Ohio

EXIHIBITIONS

"METAFABLE", Animation Screening Room SIGGRAPH 86,
Dallas, TX. (August 1986).

"Motion Studies", Animation Screening Room SIGGRAPH 85,
San Francisco, CA. (July, 1985).

"Motion Studies, Single Frames", Computer Graphics
World, (October, 1985). pp. 56.

"42nd Biennale di Venezia" La Bienale di Venezia
Venice, Italy. June '86 - August '86

Fields of Study

Major Field: Computer Graphics

TABLE OF CONTENTS

ACKNOWLEDGEMENTS ii

VITA iii

TABLE OF CONTENTS iv

LIST OF FIGURES v

CHAPTER PAGE

Chapter 1. Introduction 1
Chapter 2. The Three-dimensional Computer Environment 6
Three-dimensional Object Motion: Simple Transformations 7
Chapter 3. Developing Monolithic Objects into Rigid Characters 14
Slow-out, Slow-in (Acceleration/decelration) . 14
The Physical Laws of Motion 15
Simple Squash and Stretch 19
Object Attachment and Articulated Appendages for Characters 24
Chapter 4. Shape Change in animating Computer Characters 28
Deformations 30
Shape Interpolation 38
Growth, Fractals and Particle Systems 43
Chapter 5. Character, Environment Interaction 50
Chapter 6. Conclusion 55

References 58

LIST OF FIGURES

Figure 1.
Example data file for a cube 7

Figure 2.
Chart of two possible ease rates, showing the degrees of rotations per frame 19

Figure 3.
Bouncing ball using squash and stretch 22

Figure 4.
A curve defined by interpolation 33

Figure 5.
A curve defined by approximation 34

Figure 6.
Shape interpolation 40

Figure 7.
Shape interpolation with unexpected results 41

CHAPTER 1

Introduction

> His job as a animated character is to communicate story ideas in the most entertaining way, and just being alive is not enough [1].

Walt Disney set the standard for a character's role, and the standard for animators at the Disney studios. With development of Walt Disney's character animation into traditional animation has come an understanding and acceptance we have all grown up with on Saturday mornings. Unfortunately, for many different reasons, the traditional art form of character animation found on Saturday mornings has fallen in quality from the standards set in the past. Whether this is due to the fact that Saturday mornings have grown beyond the original purpose or, whether the children of today have new directions in entertainment, is an important study. The focus of this thesis, however is on the relationship of computer technology to character animation. Specificly it will address the advantages and constraints of developing three dimensional characters for computer animation.

With computer graphics, the tools of the animator are no longer pen and paper, but a complicated mixture of high-tech equipment and interwoven software which, when all organized properly, take care of all the drawings. Behind all the technology of computer graphics, a successful animator still must understand and implement the principles of traditional character animation.

> The general appearance of the character can be almost any design that fits the story and the overall style of its presentation, but the specifics of how he is drawn depend entirely on what business he has to do, what attitudes he must show, and what expressions he will have [1].

The same traits need to be considered when developing a computer generated character as when developing a traditional character. Although the purpose of the character remains the same in both media, the physical differences between the two are very large and far reaching. Traditional characters are visually developed through a series sketchs and drawings used in exploring different character expressions. The lines used to build up the character have a two dimensional relationship. The effect of the three dimensional environment is based on the animator's ability to perceive and draw the three dimensional perspectives views. Computer generated

characters, on the other hand, are actually constructed and are descriptions of objects in three dimensional space. The three dimensional perspective views of the character are mathematically rendered by the computer. The character or object information is stored inside the computer in various formats. Changing a character's expression involves a series of steps during which the animator manipulates the character or object information stored in the computer. In computer graphic terms, a character is constructed out of a single or combined group of objects. Each has a set of three-dimensional points and polygons, or surfaces between related points. In chapter 3 I will discuss the other three dimensional structures that can represent computer characters. These structures permit new and different character traits.

The thesis examines the different levels of complexity involved in animating computer characters. A three-dimensional computer environment and simple motion within this environment, is described. Character motion control are described in chapters 1, 2, 3. The three chapters on motion control examine: motion control of rigid, monolithic objects as characters; shape change techniques for animating characters; and the principles of

three-dimensional environment, character interaction. Examples of the character motion control being examined, are shown in the computer animation 'METAFABLE' [7].

The character's existence revolves around the information he brings to the animated story. This is true in any content no matter what techniques were used to breath life into the story. The techniques and principles used in traditional character animation are invaluable for the computer animator. Some of the principles and techniques may mislead the computer animator if the major differences in character development are not understood. The consequences of these differences go beyond the elements of creating individual characters. The emphasis shifts from the sketch, the hand, the eye and the mind to a much different set of mental and physical considerations. With computers at any level, the assigning of values to variables is a factor that is to be addressed. When producing even the simplest and smallest part of a computer animation, thousands of variables need to be set. In traditional character animation a relativity similar number of specialized decision are being made all the time. However, when designing or drawing the animation, a large number of decisions are made

subconsciously or almost instantly by simple gestures. These differences in developmental procedures point to the fact that there are differences between traditional and computer animation and in how the animator makes and implements aesthetic decisions. The shift in where and how the creative process is used is barely noticed and is far from being understood. New creative tools need to be developed and taught to the young animator, incorporating both the principles of traditional animation and the new skills and techniques involved with computers. These tools are then necessary before computer character animation can be explored to its fullest potential.

CHAPTER 2

The Three-dimensional Computer Environment

Three-dimensional computer environment consists of an imaginary three-dimensional space or world. This space or world is mathematically defined and described in the computer. Distance in this space, is defined as position on a grid of predefined units. The accepted standard this description, defined in early mathematics, is the Cartesian coordinate system. The center of coordinate system, (0,0,0) is called the origin. As taught in elementary geometry courses: the X axis runs horizontally and the Y axis runs vertically. Negative X extends to the left of the origin and positive X extends to right. Up is positive Y with negative Y going down. In three dimensional space, the Z axis advances forward and -Z recedes back in space.

Describing objects in three dimensional computer environment involves breaking the objects down into points. The computer stores this as a list of numbers containing the object point positions in three dimensional space. The way points are to be connected to make

polygons is also stored. At CGRG, object data is stored is in a single list of points and polygons. Each point has three numbers stored with it, the x,y and z positions. The polygonal structure of the object is broken up into the individual points used to define each polygon. The order of the points gives a description of how the computer needs to connect the points to form the polygon. This is much like a dot to dot drawing where the individual dots are connected by lines but in three dimensional space. Understanding object construction is important because object motion is the result of manipulating the points which construct the object. How the object data is changed shows the effect of a specific types of motion.

Three-dimensional Object Motion: Simple Transformations

Prior discussing object motion independently, one must first understand positioning an object into the three-dimensional, computer environment. The principles used in placing an object into a scene are the same as those used to move an object through the same scene. The

```
:::: cube.det ::::
Detail file (det)
8 points 6 polygons
Point   1:              1              1              1
Point   2:              1             -1              1
Point   3:             -1             -1              1
Point   4:             -1              1              1
Point   5:              1              1             -1
Point   6:             -1              1             -1
Point   7:             -1             -1             -1
Point   8:              1             -1             -1
Poly    1: 4 verts      1      2      3      4
Poly    2: 4 verts      5      6      7      8
Poly    3: 4 verts      1      5      8      2
Poly    4: 4 verts      2      8      7      3
Poly    5: 4 verts      3      7      6      4
Poly    6: 4 verts      4      6      5      1
```

Figure 1.
Example data file for a cube.

fundamentals of object placement and motion are built around the principles of object transformation.

Objects can be made larger or smaller, or scaled. They can be moved around in space, or translated. They can be turned in space, or rotated. The manipulation of an object can be performed on a copy or instance of the original. In this way we can have several instance of the same original object and manipulate each individually. Each copy can

> be given a name to assist in keeping track of the many objects in a scene. These manipulations of the object are transformations. They are based on standard mathematical techniques of coordinate geometry, trigonometry, and matrix methods. These transformations can be expressed mathematically in matrix algebra by a single entity called the transformation matrix. Several transformations of an object can be combined or concatenated into one matrix [10].

The three fundamental transformations are: translation, rotation and scaling. Each operation multiplies the point structure of the object by the proper matrix, changing the position of some or all of the points. With scaling, the simplest of the transformations, each coordinate is multiplied by a scale factor. This can be done independently to any of the axes, where the object will appear to stretch when scaled more in one axis than the others. If scaling is the same in all axes, then the object will simply appear to grow or shrink. With translation, the object coordinates are either added or subtracted by a placement value in order to move it around in space. Addition moves the object in a positive direction, while subtraction or adding negative numbers moves the object towards the negative end of the axis. Rotations are the most complex transformations. They involve a series of matrix multiplications with sines and

cosines. Each axis has its own matrix. When the object is rotated around more then one axis then all the individual matrices need to be simplified into one matrix. Rotations can be around the object's center of interest, called on-origin rotations, or around some other point in space, called off-origin rotations. An on-origin rotation is like the earth spinning on its own axis, and an off-origin rotation corresponds to the earth orbiting around the sun [10].

Discussion has centered only on the object as a single piece of data up to this point. As far as the computer is concerned this is correct. In terms of developing a three-dimensional, computer image, it is very often necessary to merge many simple objects into one. In these cases, a more complex object which is treated as a single object. In Chapter 2 the issues involved in objects with many pieces and how they can be developed into characters are described. For now, objects can be thought of as single piece elements.

We perceive motion continually through out each day. Not until the discovery of the camera and then the motion pictures could we reproduce the same motion over and over

again. The computer can easily reproduce identical motion. The output of the motion, remains the major problem in computer graphics. The motion can be displayed on monitors but is limited to where the monitor can be viewed. One answer to the output problem is to transfer the images to film or video. Similar to traditional animation, film moves through the film projector at a rate 24 frames per a second. When viewing a film, 24 single images are shown every second. With video, the tape moves at a rate of 30 frames (single images) a second. To expand into three-dimensional computer environment, think of a cube that rotates completely around every two seconds. Breaking the full rotation into incremental stages so that each successive stage depicts a small advance in the direction of the rotation. In order to achieve a full revolution in two seconds 48 cube stages must be successively displayed. At the half way point in the rotation the cube has moved a half circle or 180 degrees. With film there is 24 frames in every second. The cube as traveled half way around the circle using 24 individual stages. Each stage is one 24th of 180 degrees. A half circle divided by 24 is 7.5 degrees. To show a cube rotate once around in two seconds, a small advance of 7.5 degrees

is made for every frame. A total of 48 separate images are combined to generate the complete rotation.

Motion is the action or process of changing place or position in space. To rotate the cube 360 degrees in two seconds, the act is to rotate the cube 7.5 degrees. This act is repeated 48 times, once every 48th of a second. The key word in computer graphics is "process". For the computer each set of actions is divided into an individual process which represent a single act. The repetitive execution of a process already predefined is ideal for and is one of the major strengths of a computer. In the case of a complex motion, where the object may spin in several directions simultaneously while also moving through the space and shrinking, a set of processes are linked together and repeated as needed. Many of the different processes for moving an object have been discussed however, an object is not the only element that can be moved in a three-dimensional computer environment. In traditional, animation the motion in the three dimensional was an illusion created by the animator. In computer graphics, the third dimensional is a mathematical reality, but the principles of camera or eye motion can create a similar illusion. The eye motion is not limited by any

sense of real space. It can be placed in any position and even point in any possible direction. This includes placing the eye inside an object and having it move out through the object. The example of the rotating cube can be used now to show the illusion of motion created by moving the eye. If the cube stays ridged in the center of the space and the eye moves around the cube at the rate of 7.5 degrees a frame (aways pointing at the center of the cube) it will be impossible to distinguish which way the motion was generated. At any time the computer animator can have the eye and any number of object moving at once. This quickly develops into a complex list of processes which the computer must follow.

CHAPTER 3

Developing Monolithic Objects into Rigid Characters

Slow-in, Slow-out (accelerations/decelerations)

The movement of any monolithic object can be very detailed and calculated, resulting in a whole set of interesting visual effects. Yet, the object will continue to appear as a rigid object drifting through space. The process of bringing life to the object and developing it into a character requires an examination into the principles of animation in general. Many of the principles relate to the entire piece of work, but if the focus is limited to developing the ridged object into a character, three major principles need to be considered. Our understanding of motion is based on natural phenomenon, therefore careful study of our natural surroundings will provide insight into how to develop life within a character. The first principle is Slow in - Slow Out. All motion speeds up gradually and slows down gradually. To understand this simply try any motion, say a hand wave. It

is impossible for one to move the hand with a instant start or stop. There will always be some fraction of acceleration or deceleration into and out of movements. In computer graphics, this problem is addressed with the terms ease-ins and ease-outs. The next principle is that which applies to arcs. Most motions are not straight but follow some type of curve. Both of these animation principles are based on the forces that govern nature, the laws of physics.

The Physical Laws of Motion

During the late 1600's in Enland, Isaac Newton established the original ground work explaining the laws of motion. His three laws of motion are as stated;

Newton's Laws of Motion

1) Every object continues in a state of rest, or of uniform motion in a straight line, unless it is compelled to change that state by forces acting upon it.

2) The general case of the first law; ' F = ma ', the force of an object is equal to its mass times its acceleration.

3) To every action there is always opposed an equal reaction; or, the mutual actions of two bo-

dies upon each other are always equal, and directed to contrary parts.

Newton's laws of motion can be extended into a set of of simple concepts. Many rules can be set up to explain a wide range of possible motions. A description of some situations are detailed in " Timing and Movement in Three Dimensional Computer Animation" [11].

1) An object moving from on point to another accelerates from rest and decelerates to a stop. The changes in speed can vary depending on the size of forces acting upon the object.

2) In order to move in the opposite direction, an object must slow down before turning and then speed up again.

3) The weight of an object is described by the amount of force, gravitational or otherwise, required to move or stop it.

4) It takes a large force to move a heavy object and a proportionally smaller force to move a light object.

5) A thrown object traverses a parabolic path through the air. Due to gravity, its vertical speed diminishes to zero and then increases again as it falls toward the ground. Its forward movement remains fairly constant, depending upon the object's surface area then air friction slows it down.

6) It is actually the center of mass of an object which travels along a parabola. This is important to understand with irregularly shaped objects and objects which are rotating as they move through the air.

7) Friction causes an object to slow down. Friction is the force exerted by ground or surface which prevents gravity from being the controlling force.

Each of the seven rules have some influence on the degree of use when applying the animation principles of slow-ins, slow-outs and arcs. It is important to remember that in computer graphics, like traditional animation, there are no laws of physics, only laws and rules impose upon the scene. One object may move in an imaginary world where gravity acts at only a fraction of the normal. Another object may act like it's in a world where gravity is several times stronger. When the object is transformed

into a character, the object must move in a way in which it appears to have more life. Objects or characters that have life are forced to follow the rules of physics. It is important to apply both Slow-ins, Slow-outs and arcs when moving the object. A successful animator knows the right degree to which the true laws and the imaginary ones are applied in the process of moving the object.

Returning to three-dimensional, object motion principles as discused in Chapter 1, it is important to intergrate animation principles with knowledge of three dimensional object motion. With slow-ins and slow-outs, the important element is the slow-ins and slow-outs act as speed changes and do not affect the direction of the planned motion. Consider again the example of the rotating cube. The cube rotates 360 degrees in two seconds. In the example presented earlier, the cube rotated at a rate of 7.5 degrees each frame. Incorporating slow-ins and slow-outs, different degrees of rotation per frame would occur. The rate is no longer 7.5 degrees per frame but rather starts out and ends up with degree shifts of small amounts with larger shifts near the middle of the action. Several possible examples are shown in Figure 2. Notice how the acceleration and deceleration rates vary depending on the

force on slow-in, slow-out. No extra motion techniques are needed beyond transformation matrices, which can be used as if the motion was constant. In the case of arcs, more then simply the rate is changed. The simplest case where a object is thrown into the air, mathematical expressions called parametric equations, can be used to find the x,y values. With arcs or parabolic curves, the vertical and horizontal directions have to be addressed independently. For both directions, slow-ins and slow-outs, need to be applied. With the vertical direction the forces of gravity becomes a factor. Using slow-ins and slow-outs, gravity can be approximated. The horizontal direction uses slow-ins and slow-outs, at either end of the path [11]. When parabolic paths are projected into three dimensional, space the mathematical equations necessary to describe the motion can become very complex. After all the equations are broken down and solved, the same three dimensional matrix transformations are used to calculate the object's position.

Simple Squash and Stretch

frame	slow	fast	frame	slow	fast
frm 1	0.00	0.00	frm 2	5.73	0.40
frm 3	5.97	1.20	frm 4	6.20	2.00
frm 5	6.42	2.78	frm 6	6.64	3.56
frm 7	6.84	4.32	frm 8	7.04	5.06
frm 9	7.22	5.78	frm 10	7.40	6.47
frm 11	7.57	7.13	frm 12	7.72	7.76
frm 13	7.87	8.36	frm 14	8.00	8.92
frm 15	8.13	9.44	frm 16	8.24	9.91
frm 17	8.34	10.35	frm 18	8.43	10.73
frm 19	8.50	11.07	frm 20	8.57	11.36
frm 21	8.62	11.60	frm 22	8.67	11.78
frm 23	8.70	11.92	frm 24	8.71	12.00
frm 25	8.72	12.02	frm 26	8.71	12.00
frm 27	8.70	11.92	frm 28	8.67	11.78
frm 29	8.62	11.60	frm 30	8.57	11.36
frm 31	8.50	11.07	frm 32	8.43	10.73
frm 33	8.34	10.30	frm 34	8.24	9.91
frm 35	8.13	9.44	frm 36	8.00	8.92
frm 37	7.87	8.36	frm 38	7.72	7.76
frm 39	7.57	7.13	frm 40	7.40	6.47
frm 41	7.22	5.78	frm 42	7.04	5.06
frm 43	6.84	4.32	frm 44	6.64	3.56
frm 45	6.42	2.78	frm 46	6.20	2.00
frm 47	5.97	1.20	frm 48	5.73	0.40

Figure 2.
Chart of two possible ease rates,
showing the degrees of rotations per frame.

The third principle to consider in the object to character transition is "squash and stretch". "Squash and stretch", is the first and most important principle of traditional animation, for Thomas and Johnston, leading animators for the original Walt Disnay Studios. A key element in animating life, is that living objects rarely retain a rigid structure. "Anything composed of living flesh, no matter how bony, will show considerable movement within its shape in progressing through an action." [1] The impression of surface elasticity can be created by having the object's outer form change it's shape. Exaggerating the elasticity of an object directs attention towards the fact that the object is in control of its own shape and motion. In computer graphics there are many ways to change the surface structure of an object. Many are very difficult to specify to the computer. The most basic technique involves scaling one or more of the axis of an object. Bouncing a rubber ball up and down provides us with a good example of how adding "squash and stretch" can add extra life to the object, in this case the round ball. In addition, the "squash and stretch" can be achieved simply by changing the scale of the ball along the x,y and z axis at different rates. If the ball

bounced using the regular laws of gravity and physics, it would squash only slightly depending on how flexible the ball was. With the proper slow-ins and slow-outs calculated, the ball would appear to look realistic while bouncing, yet it would not seem to have any life of its own. Exaggerating its elasticity by squashing and stretching, would be the first step in bringing the ball to life. What is meant by squashing and stretching the ball ? A simple squash, stepping on the ball, can be thought of as scaling the ball greater in x and z or shrinking the y scale. The same process can also thought of as stretching, pulling the ball out in the x and z direction. Relating these ideas to the bouncing ball, the ball should squash when in contact with the floor and then stretch up when it rebounds off the floor. If the animator using the same slow-ins and slow-outs as before, the bouncing will look awkward. The "right look" can be achieved by adjusting the eases beyond the true values expected from nature to a set of values that work with squash and stretch and help snap life into the ball. The reason "squash and stretch" by scaling is considered to be a direct approach is that it involves the same transformation principles as any linear motion or

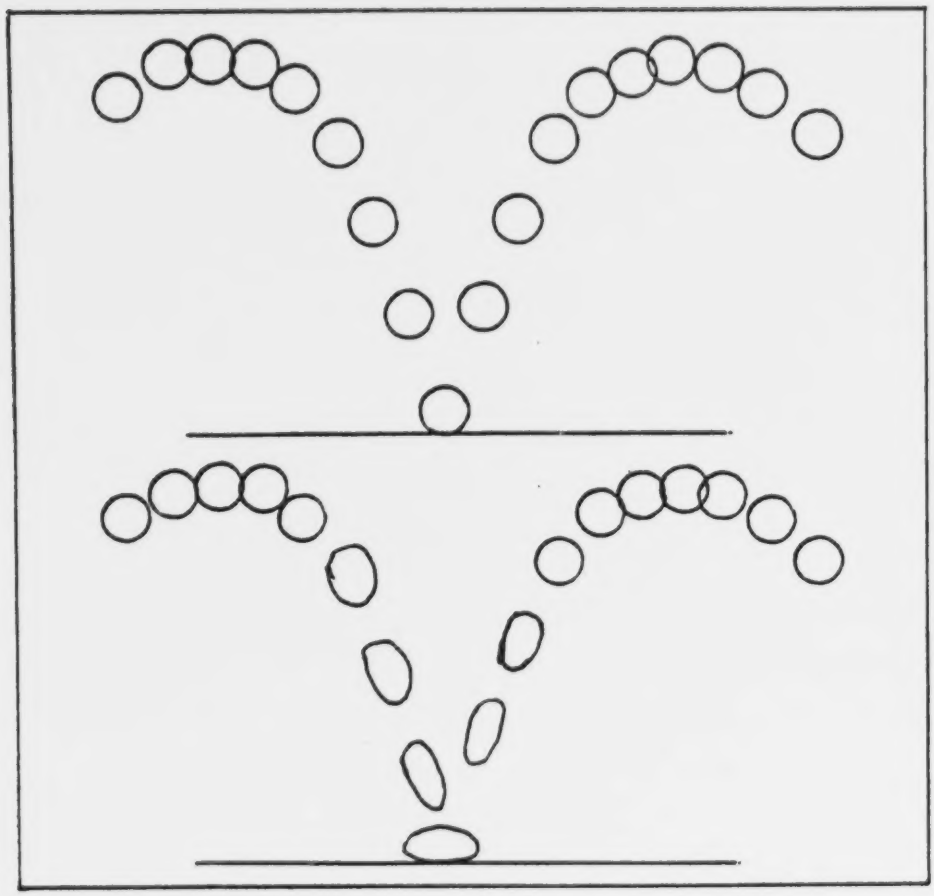

Figure 3.
Bouncing ball using Squash and Stretch.

rotation. When scaling occurs in only one or two axes, the transformation matrix needs only to multiply the correct

x,y or z coordinates by the given scalar factor.

Object Attachment and Articulated Appendages for Characters

In exploring the techniques of transforming objects into characters, the compositing of several objects into one character might be examined. A living object is almost always built up of many separate elements which all interact in one way or another with each other. It is this element to element interaction that provides the detail of motion that breaths life into objects. It is also the complexity that makes programming such motion one of the most difficult tasks in any computer animation. When each part is treated as a separate object, the principles of motion are the same as if working with a single object. Each object must be kept track of, however moving the separate parts and making them look like they are really all part of a single character is now problematic. To do this, computer animators attach the parts to one or more of the main objects. This is where the difficulty arises. There are countless ways to move the parts as a whole, or

groups of parts, or even a single part. In a computer environment, the possibilities of how these parts are attached and move in relation with each other is not limited by physics and gravity. Parts can be attached and effect each other's motion and yet not be physically touching.

It is not necessary in all cases to have flexible parts. There are certain situations where a fixed attachment adds the right amount of detail to add life. In this case, the object data is combined merging the objects into one object which moves rigidly as a whole. Having some parts fixed and others flexible is also possible.

It is helpful to remember that all the earlier motion and animation principles apply to the individual parts, as well as to the whole character. It is still important to include slow-ins, slow-outs and arcs to the motion of the appendages of your character.

In order to maintain a relationship between all the character's objects, while independent parts are manipulated, an hierarchical object structure must be established.

> A hierarchical structure refers to the way the manipulation of one object effects another associated object. These hierarchies can be set up in a "tree" structure. All the branches and leaves of the tree are attached to the trunk. Moving the trunk of a tree effects the placement of all the objects associated with that tree. Moving a leaf in that tree does not effect the placement of the trunk or other leaves. Objects in a hierarchical structure can be animated by rotating, placing and scaling the "limbs". By specifying the order of the objects in the hierarchical structure, the animator can maintain a relationship of the character's parts as well as independent control of the desired appendages. [5]

If the number of joints grows to more then a few, the complexity of control becomes exponential. With simple characters, the animator can keep track of when and how much each appendages should move or rotate from frame to frame. When thinking of building characters it is easy to think in terms of animal or human like characters. Any walking creature has many appendages that all relate to each other, and would be very unreasonable to control manually. Two different techniques are being researched at CGRG to solve this problem. Both have intergrated high level mathematics into higher level programming languages. David Zeltzer, worked on the principles of forward kinematics at The Ohio State University. The control of the appendage joints originates from the base of the limb and works outward through the limb depending on the

rotation restraints on the given joints. [13] On the other hand, Michel Girard, also at The Ohio State University, is working with principles of Pseudo inverse Jacobian Matrix mathematics to solve the same problems. With Pseudo inverse Jacobian control, an end-effector or limb end point is given. The intermediate joint angles are solved back to the base of the limb. [6]

An additional solution can be achieved by a process called "Rotoscoping". Rotoscoping is the technique of photographing the chosen walk cycle at the chosen perspective then using the images as position and rotation reference for various stages of the the walk cycle. By using images as reference points, the motion can be recreated. Unlike the mathematical solutions, the character's walk cycle generated by rotoscoping can only be viewed from the angle and perspective used in filming the original walk cycle. Walk cycles, generated with math principles, can be viewed from any angle the animator selects.

CHAPTER 4

Shape Change in animating Computer Characters

One of most promising techniques for animating computer-generated, characters is shape change, or character modification. Changing the relationship between the different parts of the character in such a way that they communicate the story is the purpose of computer animation. The idea of change is divided into two major principles, the dynamics of change and the nature of change. Each principle is closely tied to each other are an individually important in the process of shape change in computer animation. The dynamics of change involves the rate of change; and the nature of change involves the process of change. The traditional animation principle of slow-ins and slow-outs, are the most widely used techniques in controlling the dynamics of change. The rate at which one character changes from one form into another can be the key to communicating emotion or personality through shape change. Both principles are closely interdependent upon the success of the other. The techniques used to control the nature of the change are

are much more complex than the relativity simple techniques of slow-ins and slow-outs, used for the dynamics of change.

With computer animation the nature of change is the actual process used, as well as how the object's physical form has changed. In addition to the traditional animation principle of "squash and stretch", which is one of the most basic forms of shape change, there are two other traditional animation principles that apply to shape change within computer animation. The first is, follow through and overlapping action and the second is secondary action. When different shape change techniques are used on the same character at different rates and on different parts, it has the visual affect of making the character look like it has the ability of controlling several body parts at the same time. The nature of change, or shape change in computer animation can be subdivided into three general techniques: modification of the original object into a temporary form by deformation; the transition from one object form into a second object form; and the creation of a completely new original form generated each instance.

Deformations

Deformation, the first technique, modifies the original character or object mathematically by one or more of any predefined functions into a temporary form. It is this temporary form which is used as the character at any given instance during the animation. By doing this the original character form is not lost or distroyed by accident.

> Deformations allow this user to treat a solid as if it were constructed from a special type of topological putty or clay, which may be bent, twisted, tappered, compressed, expanded, and otherwise transformed repeatedly into a final shape. [3]

Above, Alan Barr talks about the special clay like quality the object can take on when being deformed or transformed into a new object. With deformation, like clay, the surface topology may change or remain the same. Unlike clay the addition or removal of the surface from an object in computer graphics is not a trivial matter. To avoid this problem, deformations which always preserve object surface topology are examined. A closer examinations of a object data file will be helpful in undersanding what happens when an object is deformed and preserves its

surface topology. As discussed earlier, object data is stored as a list of points and polygons. The surface topology of an object is defined as its surface structure or the object's polygonal structure. This means that any deformation, that preserves the surface topology of an object, changes the locations of some or all of the points but does not affect the polygonal structure at all.

A general deformation of three dimensional solid is a mathematical function F which explicitly modifies the global coordinates of points space. Points in the undeformed or original object are called (small) x, while points in the deformed object are called (large) X. Mathematically, this is represented by the equation [3]

[equation 3.1] $$X = F(x)$$

The basis to all possible deformations is that every point (x) in the object is sent through the specific function (F), replacing the the point (x) that was input with the same or different value (X) depending on how weighting factor effects the original point. The weighting factor is the term used to describe the set of (smaller) functions f, and primitive algorithms that are built into the main function (F). The functions f, and algorithms

incorporate different control information. Relative position, points values and timing are possible variables used to deform the object. Any given function F, performs a specific deformation, deformations can be categorized into several groups depending on the different types of results achieved. Objects can be bent, twisted, tappered and or warped depending on the type of functions used. A closer look into any of the deformation groups reveals additional levels of function control. These additional levels can be generalized into the group of functions that make up the weighting factors.

A simple scaling function (f), is an example of a weighting factor used in character deformation at the lowest level. If we use the equation 3.2

[equation 3.2]
$$f(x) = 2 * x, \quad x > 0$$
$$f(x) = x, \quad x <= 0$$

[equation 3.3]
$$X = F(x) = \begin{cases} f(x) = 2 * x, & x > 0 \\ f(x) = x, & x <= 0 \end{cases}$$

as our weighting functions, all points (x) in the original object that have value greater then zero, will be scaled by two and all the other points will remain the same. This is a simple case and may do little to add life or personality to our character. The complete deformation function is shown with equation 3.3.

To add more control and flexibility to the weighting functions, spline surfaces and three dimensional spline curves can be incorporated. There are a variety of ways in which to define a spline curve. They can be broadly classified as being based on "interpolation" or on "approximation". In both cases one begins by specifying a sequence of points called control points. In the case of interpolation the curve is required to pass through the control points in the order specificed. For those techniques based on approximation the curve is required to pass "near" the control points, the definition of near depends on the particular technique used. [4] With the

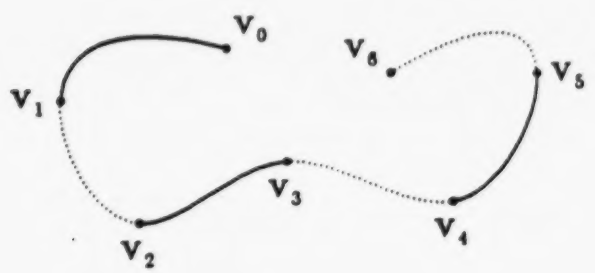

Figure 4.
A curve defined by interpolation.

smooth curve and spline smoothing function defined as P(u) and Sp(u), the locations of n + 1 control points p, the mathematical equation 3.4 shows how a smooth curve of u points, (a given number greater then n + 1 control points) can be generated. [2] The relatively few control points give an rough outline or approximation, hence the approximation technique, of the curve. The smooth version of the curve or surface is made up with many points added in between the control points. In equation 3.5 the weighting factor or specific function adds the value of the specific point in the spline surface to the original point position. Adding the spline value is only one

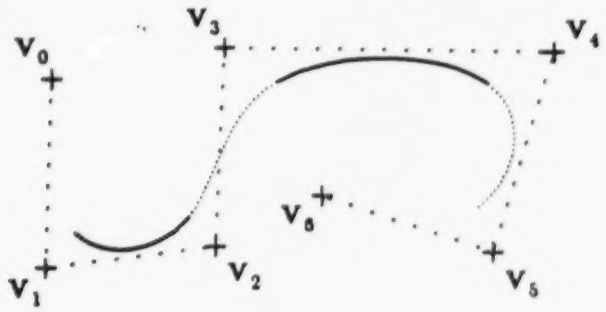

Figure 5.
A curve defined by approximation.

example of how a spline surface or curve can be used, any mathematical expression can be used. The example of adding the points has the affect of warping the object in the given axis by the spline surface. Equation 3.6 shows what the complete deformation looks like with weighting factors of both the simple scaling function used in equation 3.2 and the warping function used in equation 3.5. With this technique the weighting factor can be easyly changed by simply moving the positions of the control points, changing the curve, thus providing the animator with an interactive method of deforming an object.

control points $p = p1, p2, p3, \ldots (n + 1)$

[equation 3.4] $P(u) = p * Sp(u)$

[equation 3.5] $f(x) = x + P(u)$

[equation 3.6] $X = F(x) = \S \begin{cases} f(x) = 2 * x, & x > 0 \\ f(x) = x, & x <= 0 \\ f(x) = x + P(u) \end{cases}$

The type of deformation needed to produce any element of the character's personality or motion must be considered during the design stage of the character's development. The current technology of computer graphics leaves the animator with a group of different deformations that can be mixed and adjusted slightly to creat the desired shape. When the animator plans characters with

these restraints in mind, the character's motion will look more natural. The restraints are imposed by the principles of mathematics. When a weighting function is being used the animator is forced to think about only a couple of different variables at a time. This is true simply because an equation can not be solved if it is made up of all variables. For example, choosing to leave the z values constant forces a constraint. An animator that understands what constraints he wishes to work under and then designs his characters around them fatefully, is on the right track towards a successful animation. Each of the different deformation groups have their own set of constraints, with many overlapping other the groups. One of the more important constraints not yet discussed is the time factor. Sense the objects are moving they will be changing over time intervals. In many cases when the original object is being deformed by a spline surface the time factor becomes a weighting factor. This occurs when the original object is passed over the spline surface a given distance depending on a time variable. Other constraints that could exist are volume, arc-length, degree of tappering, twisting, etc.

A brief overview of some of the different deformation groups maybe helpful in planning a character. First the term "mapping" needs to be discussed. When spline curves or surfaces are used as weighting factors, computer animators often refer to this as mapping an object to a spline surface or curve. One of the simplest case of deformation is scaling. Scaling can occur along any of the three axes independently or together. This is often done on top of other deformations or shape changes and helps add exaggeration or squash and stretch. When the scaling is no longer linear along an axes the deformation evolves into what is know as tappering. Tappering has the affect of pinching the object at a certain point. Twisting occurs when the object is weighted with different sine and cosine values. This is much like holding a towel at both ends and ringing it out. Bending is a useful deformation, having many different applications. Bending can be done by several mathematical algorithms or by mapping an object to a two or three dimensional path. Spline functions are very helpful in creating smooth paths for the object to be mapped or bent to. The most generic form of deformation is called warping. With one of the techniques of warping the weighting factors can be used to pull at an object from

any given direction. Pushing and pulling on a half filled balloon produces much of the same results. With warping, spline surfaces are useful. Many background surfaces are created by mapping flat two-dimensional grids to warped spline surfaces. An undulating surface is created by changing the control points for the spline surface over given number of frames. In general, any form of deformation which does not fall under a specific deformation, is considered a warping deformation.

Shape Interpolation

At the early stages of story boarding and character design it is not always possible to remember all the different constraints involved with the various deformations. Many times specific character designs are already well planned before any contact has been developed with the computer software. This problem has been a problem for the traditional characters animators from the beginning of character animation. For traditional animators the problem is more specific but still involves design concepts of characters. In the Illusion of Life

,Straight ahead action and Pose to Pose, are discussed as one of the principles of animation. Straight ahead action is the design technique of drawing frames from a given starting point, or character pose, then continuing to change the characters position little by little depending on how the animator thinks the character should move. Deformation techniques can be considered somewhat within area of straight ahead action. In this case an original character is already designed and goes through its actions through various deformations. The traditional pose to pose technique occurs when the animator creates two separate frames or character positions then draws the in between frames connecting the two original frames. The technique which more closely resembles pose to pose, is called shape interpolation. Shape interpolation is the second of the three shape change techniques. A temporary form is created by interpolating between two independently designed objects is this case.

The first examples of the "squash and stretch" in computer animation were achieved with scaling techniques. With scaling only general body motion was possible when animating a character. More advanced squash and stretch techniques can be accomplished with shape interpolation.

To create very subtle expression change on a character, two similar objects need to be constructed with only specific differences in the places where the expressions occur. This can produce effects like smiles, wagging tails or even ripples in the character's surface.

Shape interpolation, like all present change shape techniques, forces the animator to follow a set of

Figure 6.
Shape interpolation

constraints. The basic principle of shape interpolation is that the position of every point on the first object in translated to the position of the same point on the second object. The point order for the object datafile is critical because shape interpolation assumes that the polygonal structure of both objects are the same. One of two object constructions techniques should be used to avoid any unpredictable. One is to built both objects by the same techniques, careful to check for the same polygonal structure. The other technique uses the same object, but changing it slightly with any of the deformations techniques.

Shape interpolation has been discussed as a technique for adding life and expression to a character. Characters have retained their general form and personality. Computer animation, like traditional animation, is not restrainted by any of the rules that govern the real world. Characters can magically transform from one form into another, without any concerns of mass, volume or gravity. Examples of liquids transforming into living creatures or of trees turning into people and walking away, have all been tried in traditional animation. These very same ideas can be done will shape interpolations, and

Figure 7.
Shape interpolation with unexpected results.

look more realistic because rendered mathematically correct. Adding three dimensional perspective to a complex transformation can only be handled successfully by an experience by traditional animator. Transforming one form into another completely different form is still difficult in computer animation because of the problems of keeping the point order similar for the two objects. Using well designed data these problems can be solved.

Growth, Fractals and Particle Systems

The third principle of shape change is made up of a variety of very different techniques all sharing the same principle of shape change: growth. Unlike deformation, where an original object is deformed into a temporary form, the third principle, generates a new shape representing the object, by building upon the previous form every instance a shape is generated. In this way objects have the appearance that they are growing out of themselves as they change. These growing algorithms take on many forms depending on what techniques are being used to generate the object. Each of the different techniques produce a unique effect that can be used to emphasize different elements of the character. With all the growth techniques the surface topology has to change to accommodate the objects propagation. Deformations on the other hand preserve surface topology, while surface topology can be presevered or changed with change shape interpolations depending on the desired constraints. Growth can be further divided into two types of techniques: one uses random elements as a part of the generation process and the other uses predefined

mathematical algorithms and rules.

One of the early leaders to use shape generation through growth techniques in computer animation is Yoichiro Kawaguchi. Kawaguchi's principal emphasis has been on recreating different forms in nature using algorithmic growth techniques.

> The forms of nature based on spirals and ramification are generated not through the use of object data calculated by measurement, but through the use of an algorithmic structure based on the laws of nature. Although there are a myriad of forms to shapes of nature, they are represented on the basis of one common principle, which can be expressed by means of of the same mathematical expressions.
>
> This program can creat spiral figures which are composed on a set of similar, but smaller, units, by entering the direction of growth on 3-D coordinate axes and accumulating these smaller figures in proportion to the growth rate, which is expanding in a series. In the exercise designed to simulate the growth of spiral plants, a more centrifugal structure is generated by increasing the number, the angles, and the length of the branches in a series at the growth point. [8]

The specific rules Kawaguchi uses to simulate growth create spiral like forms of horns, shells, and plants. These rules incorporate the laws of nature and some of the mathematical principles of branching. When generalized, these techniques can be used to create some very

expressive shape changes for animating a character. One of the problems the use of growth techniques, and mathematical algorithms, present is that they are not as flexible as many of the deformation techniques. This forces the animator to design characters with these limitations in mind. On the positive side, any sense of growth is very difficult to simulate with either of the other two shape changing principles. The other group of growth techniques has a wider range of possible forms that can be generated. These techniques solve many of the limitations present in the spiral growth techniques.

The growth techniques which incorporate randomness, utilize a group of controlling rule sets which use the random number as an element to add complexity and detail. There are presently two major approaches being used where randomness is a key element. Both approaches attempt to solve the problems of rendering increasing, or infinite, detail using completely different techniques. The two groups are referred to has fractals and particle systems. Particle systems objects are not constructed with polygons. They consist of collections of many particles. Fractal objects are polygonal objects. The limitation of using a polygonal object, which preserves surface

topology, is one of detail. When the object moves towards the observer, the polygons get bigger and become more noticeable. This effect can continue until the whole screen is covered by a single polygon. In real life situations an object retains a consistent level of detail as it moves closer. This consistency continues all the way to the atomic level. The terminology for the solution to this level of detail problem with polygonal objects is referred to as "database amplification". Database amplification has the effect of, "the closer you get the more it looks the same" when moving in on a fractal object [12]. With this growth technique it is possible, to some extent, create the visual impression that the detail is being retained as the object moves closer. Unlike Kawaguchi's techniques where set rules are mathematically followed, fractals are based on a formal language of symbol strings and labeled graphs which controls the rules for the object construction. Object control can be introduced by changing the grammer of the language. The grammer can be thought of as a series of production rules that can be arranged randomly or by the animator. Fractals grow and change through a combination of extending, and/or subdividing. Production rules provide

information on how the subdivision or addition occurs. It also controls what type of elements are being added. The strength behind grammer-based techniques is their ability to continually subdivide depending on the surface area being shown. As long as the polygon structure remains the fractal object retains, its solid appearance. An important element of fractal techniques which contrasts particle systems, is its solid appearance. Particle systems are useful for creating non-solid or fuzzy objects. A unique group of objects and textures can be created with fractals. (e.g. converting mountains into rocks, body parts into skin textures and trees into branches.)

Objects such as clouds, smoke or hair are all classified as "fuzzy" objects. Fuzzy objects are irregular, complex and ill-defined. They have motion properties which cannot be described by the traditional transformations used with polygonal objects. Using mass crowd control techniques with a large number of particles, particles systems can begin to represent the class of objects known as fuzzy objects. The mass of particles follows a rule set initially outlined by the animator. The crowd of particles roams around randomly. The rule set provides for the definition and general shape of particle

based objects. Randomness is used similarly with fractals techniques. Kawaguchi's growth algorithms are predefined and leave out any element of randomness. Without randomness to provide the extra detail or variety, the animator has a infinite set of design discussions to make. Reeves discusses the three basic differences between polygonal rendering techniques and particle systems.

> First, an object is represented not by a set of primitive surface elements, such as polygons or patches, that define its boundary, but as clouds of primitive particles that define its volume. Second, a particle system is not a static entity. Its particles are "born" and old particles "die". Third, an object represented by a particle system is not deterministic, since its shape and form are not completely specified. Instead, stochastic processes are used to create and change an object's shape and appearance.
>
> A particle system is a collection of many minute particles that together represent a fuzzy object. Over a period of time, particles are generated into a system, move and change from within the system, and die from the system [9].

Particle systems create objects and motion without shape. This type of shape change is very useful for growing character elements such as fur or hair. It is also very powerful in generating environmental objects such as rolling fog banks, running water or drifting smoke. The power is based within the interwoven connection between

the general object shape and its motion characteristics. All three techniques have been used in animation during the last several years. Kawaguchi has produced three films using his growth techniques. [8] They were developed in very fantastic environments, where he made many aesthetic departures from reality. On the other hand Lucasfilm Ltd. used fractals and particles systems in the Genesis Demo, (from Star Trek II: The Wrath of Khan), which attempts to simulate reality.

CHAPTER 5

Character, Environment Interaction

Up to this point, character design has been discussed in terms of the character's own space. To extend the character's ability to communicate the story's ideas, the animator needs to consider the remaining space around the character. This may include other characters or background objects. The entire three dimensional space can be thought of as the environment for the animation. In traditional animation, the design concept involves working with different layers, mainly a background area and foreground area. Often several different character layers are involved in the foreground area, or different special effects layers in the background. A variety of techniques are used by the traditional animator to photograph the multiple layers, or cels. Cel is the technical term for an individual layer of a complete frame. No matter what technique is used in the final shooting, the animator has to take into account that each cel is a two dimensional surface. In contrast, the computer animator has the flexibility of working in three dimensions during both the

design and production stages.

How does the two dimensional environment effect the design and presentation of traditional animation? To use rich and colorful imagery as background cels, the animator is forced to leave the background still. It is difficult to repaint a background for each frame. If a still background is to work visually, its needs to be a long distance view. Otherwise, the viewer will notice that the background is not moving in relation to the character as the camera moves. This solution leaves the animation lacking in mid-ground space, which is needed to provide depth cues for the illusion of three dimensional space. A series of compromises must be made from scene to scene, in order to project a feeling of three dimensional space.

The problems of mixing background stills with moving foreground action are no longer part of a three dimensional computer environment. Since all the environmental objects and characters are part of the same space, the computer will render the entire environment with the correct three dimensional perspectives. This allows the animator to move the camera around any part of the three dimensional environment without layering

concerns between the background objects and characters. With current object generation techniques, rich colorful backgrounds are very difficult if not impossible to create. The solution currently used, is to generate the background image with a computer paint system, then with traditional animation techniques, layer the animated frames in front of the background image.

Character design is entering a new phase because of the addition of the three dimensional environment. The animator is now provided with a new form of possible character, environment interaction. No longer are the character and environment separate entities. The question of how characters are introduced into the story takes on a new range of possible solutions. Still environmental elements can come alive becoming characters, or spawn other characters through various transformations. The possibilities are limitless and for the most part unexplored.

Walt Disney believed that the special effects department at the Walt Disney Studios was one of the most important areas. He felt that the understanding and animating of natural environmental elements was critical

to the success of the animation. The special effects people gathered information through a variety of different techniques, but the most effective was the use of live action film. To take advantage of the live action material, the animator would study the film carefully and then try to reproduce it with through drawing. Because it took so long to understand the natural elements like water, rain, smoke, and fire, animators became specialized working only on the natural elements. With computer graphics, research work on natural elements is also very important. The study of live action film can be useful, but the computer has proven to be an excellent tool for studying natural phenomenon. Most of the natural laws of physics can be converted into mathematical ideas which can then be programmed into simulation tests. Where it was difficult to transfer live action studies into traditional animation, it is relativity easy to convert computer simulations into computer animation. The relative ease in conversion is because the information gathered from a computer simulation never needs to leave the computer. The computer animator has the freedom to work with the actual information or exact copies of the information. Next the animator can manipulate any of the simulation results,

making aesthetic choices, in producing the effects desired. Already at this early stage in computer graphics, many different techniques are being used to simulate natural phenomenon. Techniques like fractals and particle systems lead the way in this area of research.

CHAPTER 6

Conclusion

The age of hard industrial production is rapidly moving into the age of information known as the communication revolution. Computer and related technologies are expanding at an exponential rate. Computer graphics and animation are at the heart of this revolution. It is very difficult and many times unrealistic to make any prediction into the future for computer animation. To retain any type of control of how the future unfolds, animators must struggle to understand and direct this mass of incoming information. Because animation is so closely tied to the center of all levels of communication, it is critical that the computer animators of today establish some guidelines.

As technology continues to expand, the working environment of the computer animator is becoming more and more complicated, forcing the animator to adapt to new and ever changing environments. Much of this stems from the fact that there are no industry guidelines. Research is being focused in all directions with very few connecting

elements. Individual companies are continually producing their own hardware and new software to run with these new machines. At CGRG the animator is involved with several different systems. He must be able to understand each system independently, and also how to interlink each system, which at times, is a very difficult task. At the current rate that technology is increasing it is only a matter of several years before a whole new set of machines will be available with new and expanded capabilities. Again, the animator will be forced to develop new skills.

Interfacing between different animation software is one of the largest problems for any computer animator. It is rare when output from different programs can be easily intergrated with that of another. If computer animation is to go forward with continuing success, guidelines must be set and followed, checking the compatibitity of one program to another. A large percentage of the development time for an animation using complex character motion is spent writing small programs that simply change the output of one program so it fits the input of another program. The animator should never have to deal with this type of problem. Many techniques have been introduced around the country to aide development of computer animation.

unfortunately, most of the software that uses these techniques cannot be intergrated. They only work on specific systems or computers. Many times the research has not been completed, so the software only works in specific cases. The first step in solving this problem is to develop an industry standard. With an industry standard, complete packages or tool sets could be marketed. For a computer animator to work freely he needs a complete set of tools or character modification programs. It is important that all the different programs can be mixed and adjusted to fit the animator's needs.

With the possibilities of what computer animation will become in the future, it is evident that computer graphics and animation is still in its infancy. Whether computer graphics will develop into a new medium is just beginning to be addressed within the computer graphics industry. It is important now that some new set of computer animation principles be developed. It is obvious that many of the traditional animation principles need to be incorporated, but it is also apparent that many other disciplines need to become involved. Areas like dance, cinema, design and mathematics all lend themselves to computer animation.

References

1. , Frank Thomas and Johnston, Ollie I. Disney Animation The Illusion of LIfe. Abbeville Press, New York, 1984.

2. , William M Newman and Sproull, Robert F. Principles of Interactive Computer Graphics. McGraw-Hill Inc., 1979.

3. Barr, Alan H. "Global and Local Deformations of Solid Primitives". Proceedings: SIGGRAPH 1984 18, 3 (July, 1984), pp. 21-30.

4. Bartels, Richard H, Beatty, John C, and Barsky, Brian A. "The Use of B-Spline and Beta-Splines to Model Freeform Curves and Surfaces". Course Notes: SIGGRAPH 1985 17, (July, 1985).

5. Bearle, Susan L Van. Computer-Generated, Three-Dimensional Character Animation. unpublished, Computer Graphics Research Group, The Ohio State University, 1983.

6. Girard, Michael. "Computational Modeling for the Computer Animation of Legged Figures". Proceedings: SIGGRAPH 1985 19, 3 (July, 1985).

7. Hutchinson, Thomas L and , Willaim Sadler. METAFABLE 1986.

8. Kawaguchi, Yoichiro. "A Morphological Study of the Form of Nature". Proceedings: SIGGRAPH 1982 16, 3 (July, 1982), pp. 223-231.

9. Reeves, William T. "Particle Systems - A technique for Modeling a Class of Fuzzy Objects". Proceedings: SIGGRAPH 1983 17, 3 (July, 1983), pp. 359-375.

10. Sachter, Judy E. The Basic Concepts of Three-Dimensional Computer Graphics for Artist. unpublished, Computer Graphics Research Group, The Ohio State University , 1982.

11. Schweppe, Marla K. Timing and Movement in Three-Dimensional Computer Animation. unpublished, Computer Graphics Research Group, The Ohio State University, 1984 .

12. Smith, Alvy Ray. "Plants, Fractals, and Formal Languages". Proceedings: SIGGRAPH 1984 18, 3 (July, 1984), pp. 1-6.

13. Zeltzer, David. "Motor Control Techniques for Figure Animation". IEEE Computer Graphics and Application 2, 9 (November, 1982).